BOLT BERANEK AND NEWMAN INC

CONSULTING · DEVELOPMENT · RESEARCH

AFCRL-68-0054

AD666443

HUMAN FACTORS AND THE DESIGN OF TIME SHARING COMPUTER SYSTEMS

R. S. Nickerson
J. I. Elkind
J. R. Carbonell

Contract No. F19628-68-C-0125
Project No. 8668
Task No. 866800
Work Unit No. 86680001

Scientific Report No. 2

This research was sponsored by the Advanced Research Projects Agency under Order No. 627

15 November 1967

Contract Monitor: Hans Zschirnt
Data Sciences Laboratory

Prepared for:

AIR FORCE CAMBRIDGE RESEARCH LABORATORIES
OFFICE OF AEROSPACE RESEARCH
UNITED STATES AIR FORCE
BEDFORD, MASSACHUSETTS

AFCRL-68-0054

HUMAN FACTORS AND THE DESIGN OF TIME SHARING COMPUTER SYSTEMS

R. S. Nickerson
J. I. Elkind
J. R. Carbonell

Contract No. F19628-68-C-0125
Project No. 8668
Task No. 866800
Work Unit No. 86680001

Scientific Report No. 2

This research was sponsored by the Advanced Research Projects Agency under Order No. 627

15 November 1967

Contract Monitor: Hans Zschirnt
Data Sciences Laboratory

Prepared for:

AIR FORCE CAMBRIDGE RESEARCH LABORATORIES
OFFICE OF AEROSPACE RESEARCH
UNITED STATES AIR FORCE
BEDFORD, MASSACHUSETTS

Abstract

The advent of computer time sharing poses an extraordinary challenge to human factors research during the next decade. Before time sharing, two facts combined to de-emphasize the importance of human factors considerations in the design of computer systems: (1) the cost of the computer's time was exorbitantly high relative to the cost of users' time, and (2) the users constituted a select, highly skilled and highly motivated group of specialists. Two of the promises of time sharing, however, are (1) a drastic reduction in the cost of computer time to the individual user, and (2) the large scale availability of computer facilities to individuals untrained in any areas of computer technology. Human factors considerations then become important both for economic and psychological reasons. This paper briefly notes what a few of these considerations are.

The rate of growth of computer technology during the last two decades has been nothing short of phenomenal. Progress has not, however, been completely smooth and continuous. It has been marked by a number of innovations that have provided impetus and sometimes new directions to the field. Time sharing is considered by many to be among the most significant of these innovations, and one that has indeed provided both thrust and direction to subsequent developments.

The original idea of using a single computer to service several users simultaneously has been credited to Strachey (Fano and Corbato, 1966; see Strachey, 1959). In the conventional batch processing mode of operation, several hours commonly elapse between the time at which a problem is submitted to a computer center and that at which it can be run and an indication of the presence of errors obtained. Moreover, several cycles of program submission, attempted execution, error correction, and resubmission are often required before a program is completely debugged. The entire process can easily run into several days. In the time-sharing system that was proposed, users would be able to interact continuously with the computer, running their programs, obtaining almost immediate information concerning the presence of errors, and correcting them on the spot.

The first time-sharing system was actually put into operation early in this decade. Subsequently, several systems have been built, many of which are currently operating. Such systems, including operating software, can now be purchased from a number of computer manufacturers. A few organizations are providing access to time-sharing systems on a commercial basis. Projects are under way to develop large new systems which contain much more sophisticated and powerful hardware and software components than are currently in use. Consideration is now being given to the possibility of forming a network of time-sharing systems by connecting them through telephone quality data lines.

The thesis of this paper is that the emergence of computer time sharing and the resulting developments pose an extraordinary challenge to human factors specialists and point up one of the most promising areas for human factors research during the next decade -- and beyond.

To focus the discussion, we should define more precisely what we mean by a time-sharing system and identify the particular subset of such systems with which we are primarily concerned in this paper. The term time sharing has come to connote a facility in which the time of a central processor and the centralized resources of the system (e.g., core and intermediate storage media) are partialled out to several concurrent users in such a way that each of them may operate almost as though he were the only user of the machine. Within this broad definition, time-sharing systems vary greatly with respect to the generality of the problems to which they can be applied. At one end of the spectrum are systems designed for a single particular application, for example, airline passenger scheduling. In such cases, the command language is very restricted and the mode of operation is highly constrained; in short, the man-machine interaction is rigidly prescribed. In contrast are those systems that are intended for fairly general use. The objective in the latter is to provide users with a choice of versatile languages and flexible operating procedures. The nature of the man-machine interaction in these cases would be expected to vary considerably with the needs and interests of particular users. Thus a physicist, a psychologist, and an electrical engineer might all use the same system, but in rather different ways. Our concern in this paper is with the latter type of time-sharing system.

Even within the domain of this more restrictive definition, systems differ considerably with respect to many of their characteristics; however, there are a number of features that one might reasonably expect to find in any time-sharing system that purports to be a general purpose facility. Not only must it be able to service several users concurrently, but the users must be able to work in complete independence of each other. User A must not be forced to make his operating procedure contingent upon what user B happens to be doing. The users must be able to operate "on-line" and to communicate with the computer in a "conversational" manner by means of a remotely located station such as a teletype or graphics terminal, or other similar device. By "conversational" manner we mean that the two-way exchange of commands, requests, and information between the man and the computer takes place on about the same time scale as do the exchanges in an interpersonal conversation. The computer must be readily accessible so that users can get on-line for unscheduled sessions as the need arises. The emphasis must be on the provision of basic tools that may be used for constructing one's own special purpose programs and for developing a computational capability specially tailored to one's own particular needs. This means in effect that the system must make available one or more user-oriented programming languages, a package of editing and debugging tools, and a set of executive commands that gives the user access to the various system services and facilities. The system should provide the individual user with a long-term storage capability so that he can file his own programs for future use, and, ideally, it should allow him to share and exchange files with other users, should he so desire. Systems with esentially these characteristics, or a nearly inclusive subset of them, have already had a considerable effect on the way that computers are used in the science and engineering community and on the productivity of many of the scientists and engineers who have used them.

Why do we single out time sharing as the one among many significant developments within computer technology that should be of particular interest to human factors specialists? Mainly because it creates an urgency for human engineering of computer systems that previously did not exist. Before time sharing, two things tended to de-emphasize the importance of human factors considerations in the design of computer systems; (1) the great disparity between the cost of the computer's time and that of the user's time, and (2) the composition of the user community. The cost of the computer's time was so exorbitantly high relative to the cost of the time of the users, it was only reasonable to concentrate on finding ways of economizing the former -- even at the expense of the latter. Partly as a consequence of this cost disparity, the large majority of individuals who really got next to computers were specialists in one or another of the areas of computer technology. A few exceptions were to be found in people working in other areas who were bold enough -- or foolhardy enough, as one may wish to perceive it -- to try to use computers personally in order to facilitate their work. For the most part, however, the community of "hands on" users was a relatively select, highly trained, and highly motivated group, willing and able to live with consoles, languages, services and operating procedures that were anything but optimal from a human factors point of view.

Time sharing is rapidly and drastically changing the situation with respect to both of these factors. The average hourly cost for access to some time-sharing facilities is now not much greater than the hourly wages of many users, and it is continually decreasing. Hence, it becomes important, for strictly economic reasons, to provide the user with a system that allows him to make efficient use of his <u>own</u>, as opposed to the computer's, time. Secondly, and perhaps even more significantly, with time sharing it becomes feasible to make powerful computational facilities directly available, on a large scale, to individuals whose training and interests are in areas other than computer technology itself. It is this implication of time sharing -- the prospect of extensive interaction between computers and persons of various disciplines and inclinations -- which, we suggest, presents a unique challenge to human factors specialists for the years just ahead. For the majority of potential users of time-sharing systems, the degree to which the systems are carefully "human-engineered" will be among the major determinants of just how symbiotic and synergetic the man-computer interaction will be.

It would be incorrect to imply that human factors considerations have played no role in the design of time-sharing systems in the past. They have. For the most part, however, features and characteristics that have proven desirable from a human factors point of view either reflect the original intuitions and preferences of the systems' builders or they represent developments that have been more or less forced over a period of time in this rapidly evolving field. In very few cases do they represent the results of efforts of human factors specialists calculated to produce systems that are optimally designed for human use. Very few systematic or formal quantitative studies have been made of the nature of man-computer interactions in a time-sharing environment or of the factors that affect the quality or productivity of those interactions.

The human factors problems that arise are many and diverse. They range from the "knobs and dials" types of considerations that are relevant to the design of user work stations to complex and elusive issues related to such problems as the development of user-oriented conversational languages, the description and analysis of man-computer interaction, and the evolution of criteria against which to evaluate systems performance.

Programming and Command Languages

Perhaps the thing that, above all others, makes the man-computer interaction different from the interaction that occurs in other man-machine systems is the fact the former has the nature of a dialogue; it involves a two-way "conversation." In general, the man communicates data and instructions concerning what operations are to be performed on those data; the computer communicates the results of performing those operations. What are the psychological principles that should guide the design of the languages in which man-computer dialogues are to occur? Do we know enough about the general characteristics of problem solving to specify what the characteristics of a problem solving language ought to be? Is it even reasonable to think in terms of a problem solving language, or does it make more sense to think in terms of several languages, each tailored to the needs of a different class of problems or a different community of users? And, if the latter is the case, how do we go about establishing problem classes or user types? What criteria do we use to evaluate existing languages and to choose between available alternatives? To what extent is the "English-like" quality of the language a desirable feature in different applications?

Human Job Swapping and System Response Time

One of the most obvious, and perhaps important, descriptors of system performance is response time, that is, the time required for the system to complete a computation and output the results. The objective of most designers of time-sharing systems has been to minimize system response times, the assumption being that the efficiency with which problems can be solved by people will depend in large part on this factor. Very short response times are obtained, however, only at some considerable cost. Not only do they require the development of sophisticated scheduling algorithms, but they may involve severe limitations on the allowable number of simultaneous users of the system and the imposition of undesirable constraints on operating procedures. The question that arises is whether the cost of reducing response times is justified in terms of resulting increases in the productivity of users. Indeed, one may question the assumption that the shorter the system response time, the more efficiently the user will interact with the computer. Simon (1966) has in fact suggested that minimization of system response time may not be the most reasonable design objective. This suggestion is prompted by the fact that in a minimum response time system, the response time is going to be variable. That is, the goal must always be the best attainable response time given the contigencies of the moment (the load on the system, the type of computation required, etc.), but since the contingencies vary, so will the response times. Simon contends that rather than allow the response time to vary continuously over a wide range, it might be better to ensure that whenever the system could not guarantee the production of an "immediate" response, it would delay its response by a time sufficiently long that the user could give his attention to another task for the duration of the delay. That is to say, whenever an appreciable delay is unavoidable, it should arbitrarily be made long enough so that the user can make some other use of his time.

But how long must a delay be before it makes sense for the user to turn his attention to another task? Simon's contention is that a man cannot effectively swap from one simple task to another unless he can count on at least several (say ten) minutes to spend on the secondary task, and that for complex tasks he probably needs an hour or more to make the swapping worthwhile. Thus, delays that are long enough to disrupt the primary task, but too short to allow swapping to a secondary one, are delays that should be designed out of the system.

Simon's proposal raises a number of interesting questions. Do people have a minimum job swapping time? How does swapping between tasks affect performance on each task and on the combination? How does the time required to make an effective swap depend on the nature of the task? Is swapping time quantized, or is it a continuous function of task complexity? These are questions that can best be answered through controlled experimentation.

Charging Algorithms and Their Effect on System Use

We have already noted that several companies are currently providing access to time-sharing systems on a commercial basis. The charge to the customer depends, of course, on the use that he makes of the system; however, the exact way in which the charge should be computed is not a priori obvious, and in fact different companies use different charging schemes. In a given session, a user might be on-line for, say, two hours. Suppose that during that time he uses three minutes of central processor time. How should he be charged for his use of the system? Perhaps from the human factors point of view, the question is better stated as follows: how will the way that he is charged affect the way in which he tends to interact with the system? That it will have an effect seems to be a reasonable assumption. If, for example, he is charged solely on the basis of the amount of central processor time that he uses, he can afford to come to the console with only a rough idea of a program in mind, and to compose the program while on-line. He can feel free to proceed at a leisurely pace, taking time out from the console to work with a pencil and paper, to ruminate on ideas that occur to him during the session, or to attend intermittently to other things. If, on the other hand, he is charged solely on the basis of on-line time, he is under pressure to preplan and prepare the work of his on-line session in some detail. He will probably feel obliged to do as much programming and debugging as possible off-line at his desk; and he will not be at ease to allow interruptions of the on-line work session for any reason. Most commercial systems in operation today use both on-line time and central processor time in the computation of charges. The question then becomes: how does the user's strategy for interacting with the system depend on the way in which these two factors are weighted in the computation?

We only complicate the problem by recognizing that the extent to which the various system components and services are used differs greatly from session to session and from user to user. It seems only reasonable that the charge to the user should depend in some way on the proportion of the total system resources that he utilizes. But again, how should the various components and services be weighted in the computation of the user's charge; and what effect will alternative weighting schemes have on the user's work habits?

There are of course many ramifications to the problem of price setting and the supplier must concern himself with all of them. It seems likely, however, that as the competition between suppliers becomes increasingly keen, more attention will be given to the fact that different charging schemes may tend to produce different patterns of system use. The practical problem that the supplier faces is to devise an algorithm that assures him of a reasonable profit from the system and at the same time does not force patterns of interaction that are objectionable on other than economic grounds. These two objectives clearly are not independent since no supplier can hope to retain customers unless they are reasonably happy with the service, and whether customers are happy with the service will depend in part on the extent to which their preferred operating procedures are also the most economical ones. Ideally, from the user's point of view, the charging algorithm should be such that it is in his best interest to use the system in much the same way that he would if it were free.

Ease of Use and Conflicting Needs of Novices and Expert Users

It seems fairly obvious that a system that is to be made available to a heterogeneous mix of users must be "easy to use." "Easy to use" is usually translated to mean that an individual should be able to interact effectively with the system after very little off-line instruction. There is, however, a considerable difference between interacting effectively and interacting efficiently. The problem is to design a system so that a novice can use it effectively after spending a few minutes with a tutor or a manual, and can expect to learn to use it efficiently from the feedback provided by the system itself. Insofar as possible, the system should be designed so that the most efficient and powerful approaches to problems are readily discovered by the user in the process of interacting with the system. That is to say, the system should have a built-in teaching capability designed to facilitate the acquisition of that knowledge and those skills that qualify a user as an expert. (Incidentally, the general purpose time-sharing computer facility should be of greater interest to no one than to proponents of automated teaching.

Not only does it provide an ideal vehicle for the study of programmed instruction techniques, but it has the necessary built-in capability for exploiting such techniques as no other man-machine system does.)

But how does one design a system so that it satisfies <u>both</u> the needs of the novice or infrequent user and also those of the individual who uses the system more or less continuously day after day? Novices need simple procedures, self-explanatory languages, lengthy and detailed error diagnostics and considerable instructional assistance from the computer. But many of the features that are useful to beginners are perceived by experienced users as annoying frills. Experienced users invariably want embellishments that make the system more powerful but also more complicated to learn; they want abbreviated commands to minimize the input time; they want to replace lengthy error diagnostics and other computer-to-user messages with one- or two-character codes. One of the most reliable marks of an experienced user of an interactive system is the ease with which he is frustrated by system features that he feels are impairing the efficiency of the interactive session. Unnecessary delays, however short, are sources of irritation, as are the occurrence of non-informative or redundant computer outputs. This problem is not solved simply by having two versions of any particular system, one for the novice and one for the expert, since it is not really possible to partition users into just these two categories. Users represent a whole spectrum of expertness. Moreover, a particular user masters a system only slowly over a long period of time, and different users, because of their own particular needs, may acquire skill with some aspects of a system while remaining relatively unskilled with respect to others.

Operational Trade-Offs

Many of the human factors problems arise not because user preferences are unknown, but because they simply are not attainable given the present state of the art. That is to say that technological limitations or economic constraints sometimes preclude the realization of features that users quite obviously would like a system to have.

Then compromises have to be struck; trade-offs have to be made. It is important in these cases to be able to distinguish between compromises that will not seriously diminish the utility of the system, and those that will. Unfortunately, it is not always possible to conclude, on the basis of existing information, what, from the user's point of view, is an acceptable trade-off and what is not.

Consider, for example, the problem that a supplier of a time-sharing service faces in establishing subscription quotas. Among the many goals that he would like to realize are both of the following: (1) maximization of accessibility of the system to the individual subscriber, and (2) minimization of waste (i.e., idle time) of the system's resources. He wants to realize the first goal because he knows that one of the most attractive aspects of a time-sharing service for many users is the possibility of getting on-line whenever the need arises. He wants to realize the second, because the busier he keeps the system, the less it should cost the individual subscriber per unit time. It is clear, however, that these goals are somewhat opposed to each other. One can <u>guarantee</u> continuous access only by limiting the number of subscribers to that which the facility can simultaneously service, which means a considerable waste of resources except during those very rare occasions when all users are on-line at once. On the other hand, one way to keep the system maximally busy is to allow a large enough number of subscribers to insure that the system is always saturated; but this, of course, means severely limiting access for most of the users much of the time. The determination of what the compromise should be is partially an economic problem and in part a psychological one. It is economic because there is a limit to the amount of unused resources that a system that is supported by paying customers can afford. It is a psychological problem because there is a limit to the number of times that a user will tolerate being refused access to the system before rejecting it altogether. These limits are not independent of each other, of course. The amount of unused resources that a system can afford depends in part on the rate at which the service is being sold. And since, presumably, accessibility is one of the things that the subscriber is paying for, how much he is willing to pay will depend in part on how accessible the system is. Clearly, he should be willing to pay more for a system that is almost always available than for one that almost never is. But exactly how much is accessibility worth? And where is the dividing line between a compromise that is acceptable and one that is not?

It is interesting to note that the problem of accessibility is intensified by the enthusiastic acceptance of time-sharing systems by their users. Almost all of the time-sharing systems that have been brought into operation have been extremely popular with their respective user communities; that is to say, they have been heavily used. Unfortunately, some of the promised benefits of time sharing tend to disappear as the load on a system increases. The M.I.T. MAC system, for example, was so heavily loaded, at one period at least, that the modal number of simultaneous users was 29 when the system capacity was only 30 (Hastings and Scherr, 1965). This, of course, guarantees that any given user will be denied access to the system on a large percentage of the occasions that he requests it. Thus the advantage of easy access is jeopardized by the very popularity of the easy-access system. This problem is by no means unique to the MAC system; it has been encountered by several others as well.

- - - - - - - - - -

We have mentioned only a few of the human factors problems that are associated with the design and use of time-shared computer systems. There are many more. For example what types of status and control information should the user have during the on-line session? What can be done with formatting conventions, the use of special symbols, color and other coding schemes to make programs and hard copy records of work sessions easier to read? How should user work stations be designed? How should changes (improvements) in system characteristics and operating procedures be introduced in a functioning system? Are there significant commonalities in the errors that novice computer users tend to make or are beginners' errors strictly dependent on the particular systems used? What are the implications of the prospective general availability of time-sharing systems for today's training of tomorrow's scientists, engineers, and other potential users? What is the proper vocabulary for describing man-computer interaction? How are we to describe and quantify the man-computer dialogue in a useful way? What can we apply from the various attempts to describe and quantify interpersonal conversations? What are the similarities and essential differences between the latter and man-computer dialogues?

The full implications of the advent of the electronic computer for the future of society and of the position of the individual within it are impossible to assess. That they are very great, however, is hardly open to debate. Much of the speculation concerning the specific ways in which our lives will be affected by computer-related technical developments has centered on the socio-economic implications of automation, as, for example, drastic reductions in the demand for various types of human skills and labor, and a general increase in leisure time. Clearly, these implications are of great importance and involve psychological and sociological problems that are worthy of considerably more attention than they are receiving. This paper has been concerned with a different, and perhaps less widely recognized, way in which the computer may affect our lives; viz., in the provision of widely accessible computational resources for the facilitation of various intellectual tasks. "Computing utilities" is one term that has been used to encapsulate the idea, the connotation being that of a communication line to a computer system being available to anyone on a subscription basis. Economic constraints, once the major obstacles to the realization of such an objective, are rapidly being overcome. Among the problems that will be of paramount significance in the near future are those relating to the general question of how these systems should be designed for optimal human use. It is our contention that here is an exciting challenge to human factors specialists, and a potentially fruitful area for human factors activity particularly during the next few years.

The problems are complex and many of the issues are poorly understood. Few of the questions that have been raised can be answered by an afternoon of experimentation; in some cases they must be better formulated before it even makes sense to consider how they might be answered at all. But we are, after all, dealing with systems that are being developed for the express purpose of augmenting human thought processes. Given our very limited understanding of that which we are hoping to augment, it would be surprising indeed if the problems were straightforward and the approaches to their solutions clearly defined.

References

Fano, R. M. and Corbato, F. J. Time sharing on computers, in Information. San Francisco: W. H. Freeman & Co., 1966, 76-95.

Hastings, T. and Scherr, A. L. Operating statistics of the MAC time-sharing system, Project MAC Memo #MAC-M-280, Dec, 1965.

Simon, H. A. Reflections on time sharing from a user's point of view, Computer Science Research Review, Carnegie Inst. of Technology, 1966, 43-51.

Strachey, C. Time sharing in large, fast computers, in Information Processing, Proceedings of the International Conference on Information Processing, Paris, June 1959, UNESCO, Paris, 1960, 336-341.

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Bolt Beranek and Newman Inc 50 Moulton Street Cambridge, Massachusetts 02138	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE
HUMAN FACTORS AND THE DESIGN OF TIME SHARING COMPUTER SYSTEMS

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Scientific. Interim

5. AUTHOR(S) (First name, middle initial, last name)
R. S. Nickerson
J. I. Elkind
J. R. Carbonell

6. REPORT DATE 15 November 1967	7a. TOTAL NO. OF PAGES 20	7b. NO. OF REFS 4

8a. CONTRACT OR GRANT NO. ~~ARPA Order No.627~~ F19628-68-C-0125 b. PROJECT NO., Task, Work Unit Nos. 8668-00-01 c. DoD Element 6154501R d. DoD Subelement N/A	9a. ORIGINATOR'S REPORT NUMBER(S) BBN Report No. 1570 Scientific Report No. 2 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) AFCRL-68-0054

10. DISTRIBUTION STATEMENT
1. Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce, for sale to the general public.

11. SUPPLEMENTARY NOTES This research was sponsored by the Advanced Research Projects Agency	12. SPONSORING MILITARY ACTIVITY Air Force Cambridge Research Laboratories (CRB) L. G. Hanscom Field Bedford, Massachusetts 01730

13. ABSTRACT

The advent of computer time sharing poses an extraordinary challenge to human factors research during the next decade. Before time sharing, two facts combined to de-emphasize the importance of human factors considerations in the design of computer systems: (1) the cost of the computer's time was exorbitantly high relative to the cost of users' time, and (2) the users constituted a select, highly skilled and highly motivated group of specialists. Two of the promises of time sharing, however, are (1) a drastic reduction in the cost of computer time to the individual user, and (2) the large scale availability of computer facilities to individuals untrained in any areas of computer technology. Human factors considerations then become important both for economic and psychological reasons. This paper briefly notes what a few of these considerations are.

Unclassified
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Time sharing systems						
Man-computer interaction						
Human engineering						
Man-machine systems						